

Operacijski sustavi 1

[Dashboard](#) / [My courses](#) / [Preddiplomski studij](#) / [Informacijski i poslovni sustavi](#) / [OS1](#) / [Laboratorijske vježbe](#)
/ [Zadatak 1 - Prekidi i signali](#)

Zadatak 1 - Prekidi i signali

Zadatak 1 - Prekidi i signali

Vježba 1: Prekidi i signali

Zadatak

Napisati program koji omogućava obradu prekida s više razina/prioriteta, bar triju, zadanih od korisnika, s tim da se prekidi koji čekaju na obradu ne pamte, ali viša razina prioriteta prekida manju razinu.

Struktura prekidne rutine dana je sljedećim pseudokodom:

```
prekidna_rutina  /* pokreće se pojavom prekida uz zabranu daljnjih prekida */
{
    ispiši_vrijeme_ulaska_u_prekidnu_rutinu();
```

```
    odredi_uzrok_prekida, tj. indeks i;
```

```
    /* ako postoji prekid koji je prioritetniji od trenutnog posla, idi u obradu */
    ako je (i > TEKUĆI_PRIORITET) tada {
        PRIORITET[i] = TEKUĆI_PRIORITET;
        TEKUĆI_PRIORITET = i;
        omogući_prekidanje;
        obrada_prekida(i);
        zabrani_prekidanje;
        TEKUĆI_PRIORITET = PRIORITET[i];
    }
```

```
} /* povratkom se automatski dozvoljavaju prekidi (signali) koji su uzrokovali poziv ove f-je */
```

UNIX ne dozvoljava pojedinom korisniku izravno korištenje prekida procesora. Stoga prekide treba simulirati koristeći *signale* koje jezgra operacijskog sustava šalje procesima.

UPUTA

Sklopovski prekid u jednoj razini simulira se pritiskom na tipku **Ctrl-C**, čime operacijski sustav programu šalje signal SIGINT. Nakon toga operacijski sustav prekida izvođenje programa, zabranjuje daljnje prekidanje i poziva korisničku funkciju za obradu signala koja u ovom zadatku simulira prekidnu rutinu.

Korisnička funkcija za prihvatanje signala mora biti funkcija s jednim cjelobrojnim parametrom (koji se može ignorirati) i mora biti najavljena u glavnom programu naredbom:

```
sigset (SIGINT, prekidna_rutina);
```

Na početku prekidne rutine mora se odrediti uzrok prekida. Obrada prekida ne mora ništa korisno raditi, već samo treba trajati neko vrijeme. Svake sekunde obrade prekida, kao i rada glavnog programa ispisati što se radi.

Npr. ispis može izgledati:

```
Poceo osnovni program
Glavni program 1/30
Glavni program 2/30
Glavni program 3/30
Glavni program 4/30
^C
Prekidna rutina pozvana u: Thu Aug 27 14:03:09 2009

Razina prekida: 1

Poceo obradu prekida 1
Obrada prekida 1: 1/5
Obrada prekida 1: 2/5
^\\
Prekidna rutina pozvana u: Thu Aug 27 14:03:30 2009

Razina prekida: 2

Poceo obradu prekida 2
Obrada prekida 2: 1/5
Obrada prekida 2: 2/5
Obrada prekida 2: 3/5
Obrada prekida 2: 4/5
Obrada prekida 2: 5/5
Zavrsio obradu prekida 2

Obrada prekida 1: 3/5
Obrada prekida 1: 4/5
Obrada prekida 1: 5/5
Zavrsio obradu prekida 1

Glavni program 5/30
Glavni program 6/30
...
Glavni program 29/30
Glavni program 30/30
Zavrsio osnovni program
```

Za ispis vremena vidjeti primjerice `man ctime`.

Kostur rješenja može biti:

```

#include <stdio.h>
#include <signal.h>
#include <time.h>

#define N 4      /* broj razina prekida */

int PRIORITET[N];
int TEKUCI_PRIORITET;

void obrada_prekida(int j)
{
    printf ("Poceo obradu prekida %d\n", j);
    /* obrada se simulira trošenjem vremena - 10 s */
    printf ("Završio obradu prekida %d\n", j);
}

void prekidna_rutina (int sig)
{
    int i;
    time_t t;

    time(&t);
    printf("Prekidna rutina pozvana u: %s", ctime(&t));

    switch (sig) {
    case SIGINT:
        i = 1;
        break;
    case SIGQUIT:
        i = 2;
        break;
    case SIGTSTP:
        i = 3;
        break;
    default:
        return;
    }
    printf ("Razina prekida: %d\n", i);
    ...
}

int main (void)
{
    sigset (SIGINT, prekidna_rutina);
    sigset (SIGQUIT, prekidna_rutina);
    sigset (SIGTSTP, prekidna_rutina);

    printf ("Poceo osnovni program\n");
    /* troši vrijeme da se ima šta prekinuti - 10 s */
    printf ("Završio osnovni program\n");

    return 0;
}

```

NAPOMENA: Ponekad signal SIGALRM može na nekim arhitekturama uzrokovati probleme (pogotovo dok funkcija scanf čita broj s tipkovnice).

Zabrana prihvata signala (npr. za signal SIGINT) simulira se naredbom: `sighold(SIGINT);`

Omogućavanje prihvata signala simulira se naredbom: `sigrelse(SIGINT);`

Last modified: Saturday, 29 February 2020, 3:02 PM

◀ [Prekidi i signali](#)

Jump to...

[sigaction.c](#) ▶

You are logged in as Kristijan Čulina (Log out)

OS1

Get the mobile app

Helpdesk:

Za sva tehnička pitanja i pomoć oko korištenja ELF-a obratite se na: elf@foi.unizg.hr